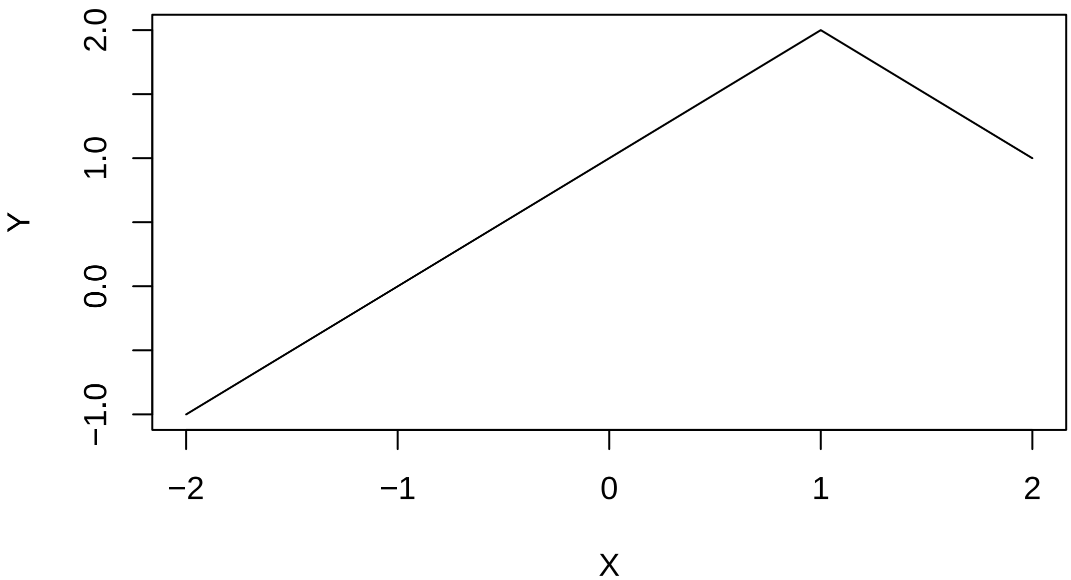
Lab 4

Zara Waheed

25th Feb 2022

Question 7.3

```
X <- -2:2
Y <- 1 + 1*X - 2*((X - 1)^2)*I(X >= 1)
plot(X, Y, type = "l")
```



Question 7.9

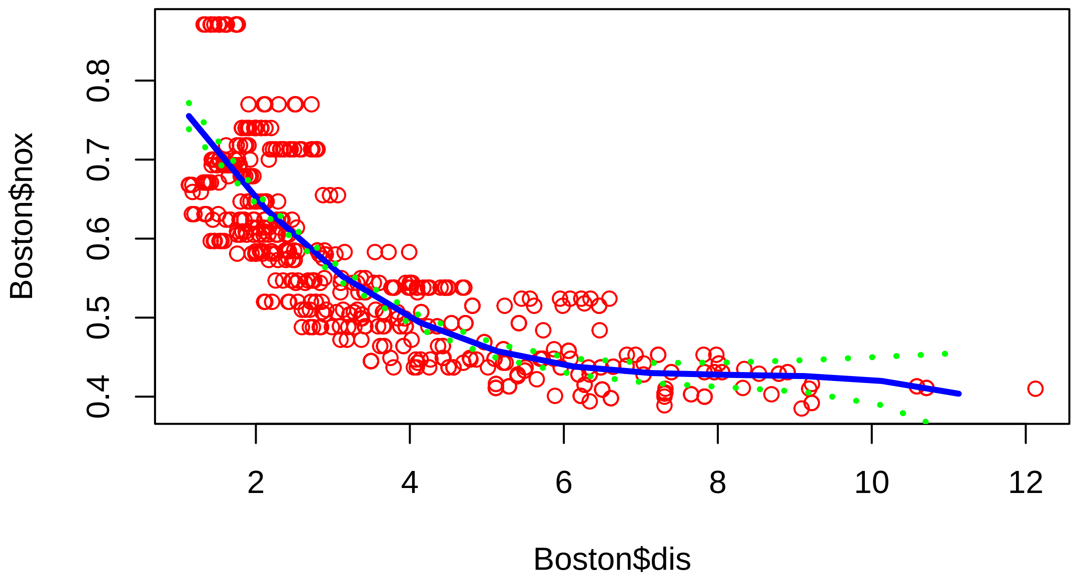
a)

```
set.seed(1)
fit7.9a <- lm(nox ~ poly(dis, 3), data = Boston)
summary(fit7.9a)
```

```
##
## Call:
## lm(formula = nox ~ poly(dis, 3), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.121130 -0.040619 -0.009738  0.023385  0.194904
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
```

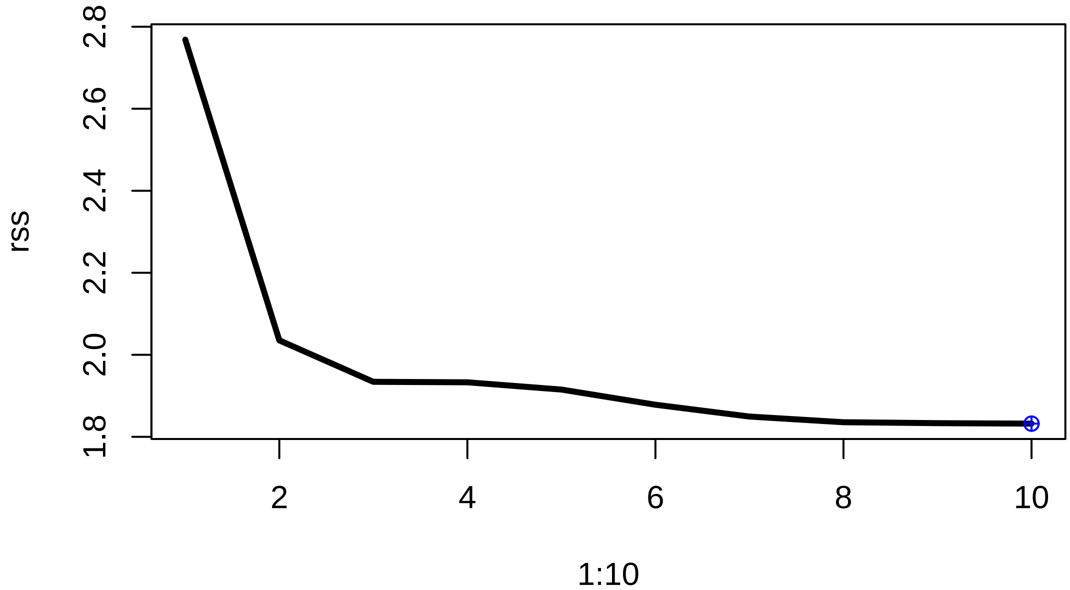
```
## (Intercept)    0.554695   0.002759 201.021  < 2e-16 ***
## poly(dis, 3)1 -2.003096   0.062071 -32.271  < 2e-16 ***
## poly(dis, 3)2  0.856330   0.062071  13.796  < 2e-16 ***
## poly(dis, 3)3 -0.318049   0.062071  -5.124 4.27e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06207 on 502 degrees of freedom
## Multiple R-squared:  0.7148, Adjusted R-squared:  0.7131
## F-statistic: 419.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
lim <- range(Boston$dis)
grid <- seq(lim[1], lim[2])
pred <- predict(fit7.9a, list(dis = grid), se = TRUE)

se <- cbind(pred$fit + 2*pred$se.fit, pred$fit - 2*pred$se.fit)

plot(Boston$dis, Boston$nox, col = "red")
lines(grid, pred$fit, col = "blue", lwd = 3)
matlines(grid, se, lwd = 3, col = "green", lty = 3)
```



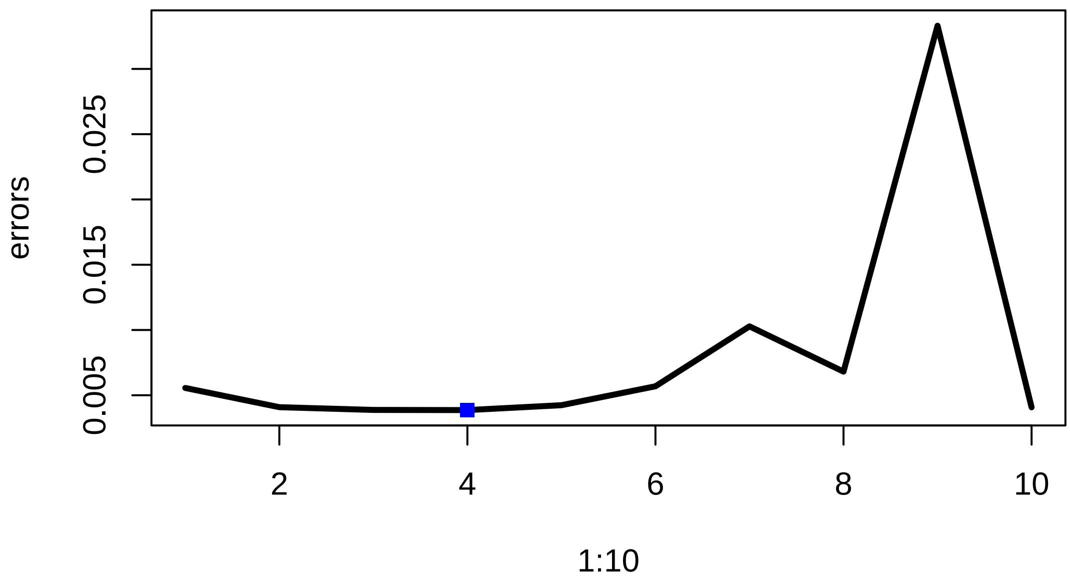
b)

```
set.seed(1)
rss <- rep(NA, 10)
for (i in 1:10){
  fit <- lm(nox ~ poly(dis, i), data = Boston)
  rss[i] <- sum(fit$residuals^2)
}
plot(1:10, rss, type = "l", lwd = 3)
points(which.min(rss), rss[which.min(rss)], col='blue',pch=10)
```



c)

```
errors <- rep(NA, 10)
for (i in 1:10) {
  fit <- glm(nox ~ poly(dis, i), data = Boston)
  errors[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
}
plot(1:10, errors, type = "l", lwd = 3)
points(which.min(errors), errors[which.min(errors)], col='blue',pch=15)
```



d)

```
summary(Boston$dis)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.130   2.100   3.207   3.795   5.188  12.127
```

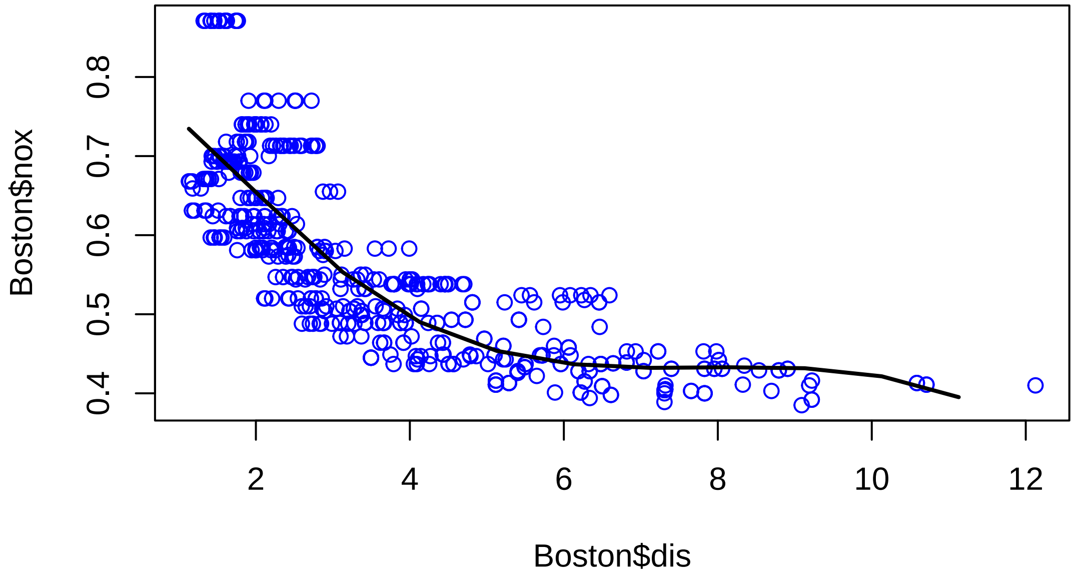
```r
fit7.9d <- lm(nox ~ bs(dis, df = 4), Boston)

summary(fit7.9d)
```

```
##
## Call:
## lm(formula = nox ~ bs(dis, df = 4), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.124622 -0.039259 -0.008514  0.020850  0.193891
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.73447    0.01460  50.306  < 2e-16 ***
## bs(dis, df = 4)1 -0.05810    0.02186  -2.658  0.00812 **
## bs(dis, df = 4)2 -0.46356    0.02366 -19.596  < 2e-16 ***
## bs(dis, df = 4)3 -0.19979    0.04311  -4.634 4.58e-06 ***
## bs(dis, df = 4)4 -0.38881    0.04551  -8.544  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06195 on 501 degrees of freedom
## Multiple R-squared:  0.7164, Adjusted R-squared:  0.7142
## F-statistic: 316.5 on 4 and 501 DF,  p-value: < 2.2e-16
```

```r
attr(bs(Boston$dis, df = 4), "knots")
```

```
##     50%
## 3.20745
```

```r
x <- seq(min(Boston$dis), max(Boston$dis))
y <- predict(fit7.9d, data.frame(dis = x))

plot(Boston$dis, Boston$nox, col = "blue")
lines(x, y, lwd = 2)
```

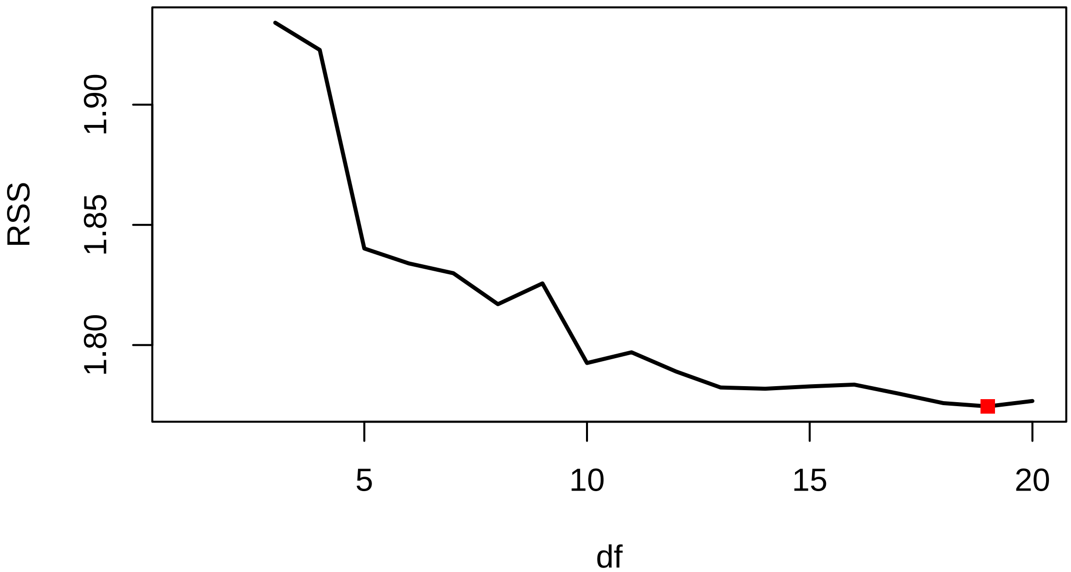


At 4 degrees of

freedom, we get the knot at 3.207

e)

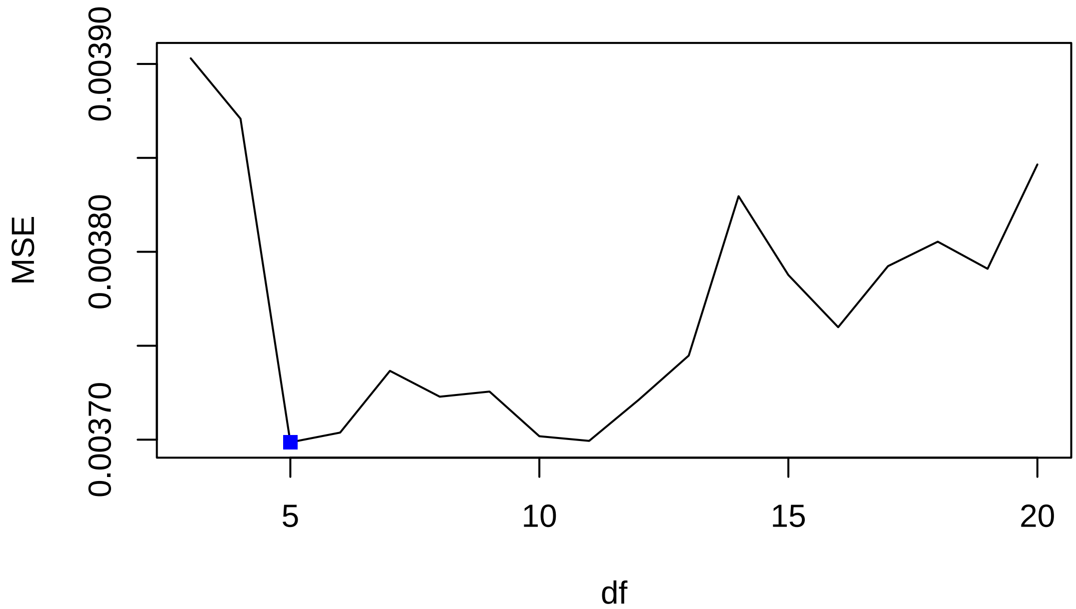
```
df_vs_rss <- c()
for (i in 3:20) {
  fit <- lm(nox ~ bs(dis, df = i), data = Boston)
  pred <- predict(fit, data.frame(dis = x))
  df_vs_rss[i] <- sum(fit$residuals^2)
}
plot(1:20, df_vs_rss, xlab = "df", ylab = "RSS", type = "l", lwd = 2)
points(which.min(df_vs_rss), df_vs_rss[which.min(df_vs_rss)], col='red',pch=15)
```



14 degrees of freedom gives us the lowest RSS value.

f)

```
set.seed(100)
cv <- rep(NA, 20)
for (i in 3:20) {
  fit <- glm(nox ~ bs(dis, df = i), data = Boston)
  cv[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
}
plot(3:20, cv[3:20], xlab = "df", ylab = "MSE", type = "l")
points(which.min(cv), cv[which.min(cv)], col = "blue", pch = 15)
```



14 degrees of freedom gives us the lowest MSE value.

Question 7.10

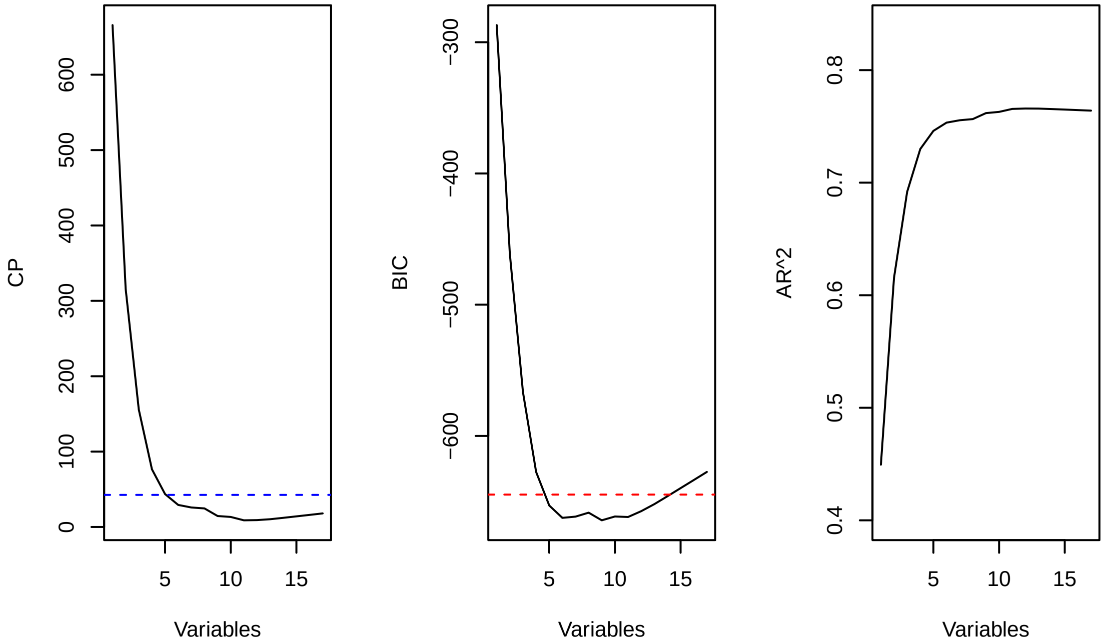
a)

```
data("College")

# Create test and train datasets
set.seed(100)
train_s <- sample(1:nrow(College), 500)
train <- College[train_s,]
test <- College[-train_s,]

fit7.10a <- regsubsets(Outstate ~ ., train, nvmax = ncol(College)-1, method = "forward")


# FSS
fss_summary <- summary(fit7.10a)
par(mfrow = c(1, 3))
plot(fss_summary$cp, xlab = "Variables", ylab = "CP", type = "l")
min.cp <- min(fss_summary$cp)
std.cp <- sd(fss_summary$cp)
abline(h = min.cp + 0.2 * std.cp, col = "blue", lty = 2)
abline(h = min.cp - 0.2 * std.cp, col = "blue", lty = 2)

# BIC
plot(fss_summary$bic, xlab = "Variables", ylab = "BIC", type='l')
min.bic <- min(fss_summary$bic)
std.bic <- sd(fss_summary$bic)
abline(h = min.bic + 0.2 * std.bic, col = "red", lty = 2)
abline(h = min.bic - 0.2 * std.bic, col = "red", lty = 2)

# Adjusted R^2
```

```r
plot(fss_summary$adjr2, xlab = "Variables", ylab = "AR^2", type = "l", ylim = c(0.4, 0.84))
max.ar2 <- max(fss_summary$ar2)
```

```
## Warning in max(fss_summary$ar2): no non-missing arguments to max; returning -Inf
```

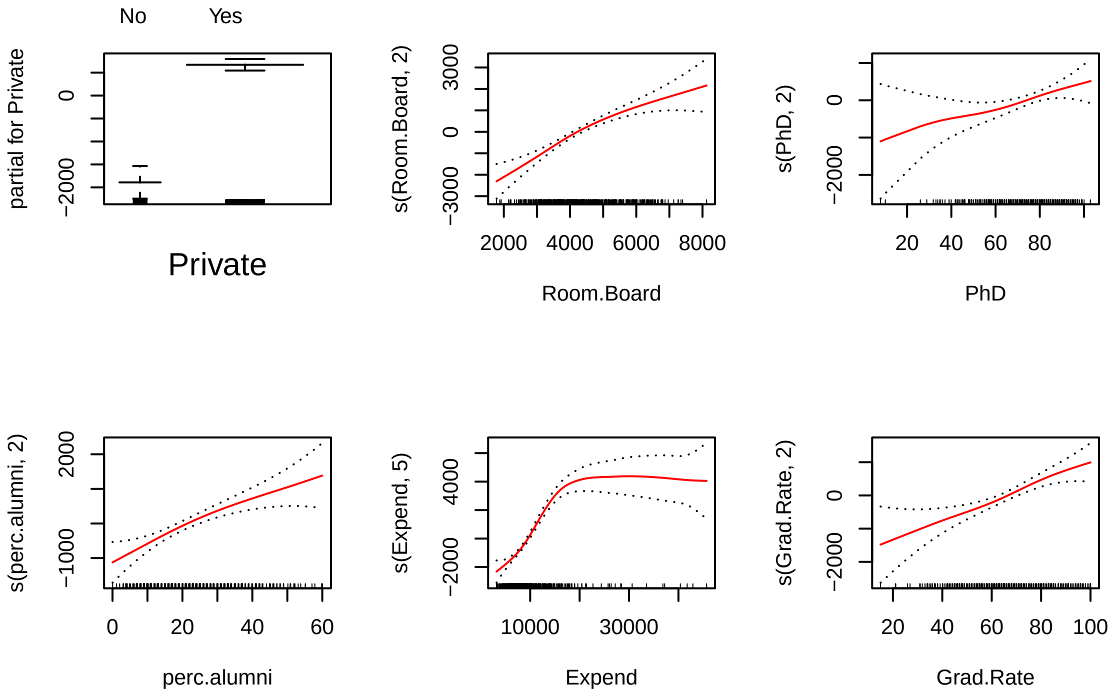
```r
sd.ar2 <- sd(fss_summary$ar2)
abline(h = max.ar2 + 0.2 * sd.ar2, col = "green", lty = 2)
abline(h = max.ar2 - 0.2 * sd.ar2, col = "green", lty = 2)
```



The model metrics do not seem to improve much after 6 predictors.

b)

```r
fit7.10b <- gam(Outstate ~ Private + s(Room.Board,2) + s(PhD,2) + s(perc.alumni,2) + s(Expend,5) + s(Gra
par(mfrow = c(2,3))
plot(fit7.10b, se = TRUE, col = "red")
```



d)

```r
summary(fit7.10a)
```

```
## Subset selection object
## Call: regsubsets.formula(Outstate ~ ., train, nvmax = ncol(College) -
##     1, method = "forward")
## 17 Variables  (and intercept)
##                Forced in Forced out
## PrivateYes         FALSE      FALSE
## Apps               FALSE      FALSE
## Accept             FALSE      FALSE
## Enroll             FALSE      FALSE
## Top10perc          FALSE      FALSE
## Top25perc          FALSE      FALSE
## F.Undergrad        FALSE      FALSE
## P.Undergrad        FALSE      FALSE
## Room.Board         FALSE      FALSE
## Books              FALSE      FALSE
## Personal           FALSE      FALSE
## PhD                FALSE      FALSE
## Terminal           FALSE      FALSE
## S.F.Ratio          FALSE      FALSE
## perc.alumni        FALSE      FALSE
## Expend             FALSE      FALSE
## Grad.Rate          FALSE      FALSE
## 1 subsets of each size up to 17
## Selection Algorithm: forward
##           PrivateYes Apps Accept Enroll Top10perc Top25perc F.Undergrad
## 1  ( 1 )  " "        " "  " "    " "    " "       " "       " "
## 2  ( 1 )  "*"        " "  " "    " "    " "       " "       " "
```

```
## 3  ( 1 )  "*"        " "  " "    " "    " "       " "       " "
## 4  ( 1 )  "*"        " "  " "    " "    " "       " "       " "
## 5  ( 1 )  "*"        " "  " "    " "    " "       " "       " "
## 6  ( 1 )  "*"        " "  " "    " "    " "       " "       " "
## 7  ( 1 )  "*"        " "  " "    " "    " "       " "       " "
## 8  ( 1 )  "*"        " "  "*"    " "    " "       " "       " "
## 9  ( 1 )  "*"        " "  "*"    "*"    " "       " "       " "
## 10  ( 1 ) "*"        "*"  "*"    "*"    " "       " "       " "
## 11  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       " "
## 12  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       " "
## 13  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       " "
## 14  ( 1 ) "*"        "*"  "*"    "*"    "*"       "*"       " "
## 15  ( 1 ) "*"        "*"  "*"    "*"    "*"       "*"       " "
## 16  ( 1 ) "*"        "*"  "*"    "*"    "*"       "*"       "*"
## 17  ( 1 ) "*"        "*"  "*"    "*"    "*"       "*"       "*"
##           P.Undergrad Room.Board Books Personal PhD Terminal S.F.Ratio
## 1  ( 1 )  " "         " "        " "   " "      " " " "      " "
## 2  ( 1 )  " "         " "        " "   " "      " " " "      " "
## 3  ( 1 )  " "         "*"        " "   " "      " " " "      " "
## 4  ( 1 )  " "         "*"        " "   " "      " " " "      " "
## 5  ( 1 )  " "         "*"        " "   " "      " " "*"      " "
## 6  ( 1 )  " "         "*"        " "   " "      " " "*"      " "
## 7  ( 1 )  " "         "*"        " "   "*"      " " "*"      " "
## 8  ( 1 )  " "         "*"        " "   "*"      " " "*"      " "
## 9  ( 1 )  " "         "*"        " "   "*"      " " "*"      " "
## 10  ( 1 ) " "         "*"        " "   "*"      " " "*"      " "
## 11  ( 1 ) " "         "*"        " "   "*"      " " "*"      " "
## 12  ( 1 ) " "         "*"        " "   "*"      " " "*"      "*"
## 13  ( 1 ) " "         "*"        "*"   "*"      " " "*"      "*"
## 14  ( 1 ) " "         "*"        "*"   "*"      " " "*"      "*"
## 15  ( 1 ) " "         "*"        "*"   "*"      "*" "*"      "*"
## 16  ( 1 ) " "         "*"        "*"   "*"      "*" "*"      "*"
## 17  ( 1 ) "*"         "*"        "*"   "*"      "*" "*"      "*"
##           perc.alumni Expend Grad.Rate
## 1  ( 1 )  " "         "*"    " "
## 2  ( 1 )  " "         "*"    " "
## 3  ( 1 )  " "         "*"    " "
## 4  ( 1 )  "*"         "*"    " "
## 5  ( 1 )  "*"         "*"    " "
## 6  ( 1 )  "*"         "*"    "*"
## 7  ( 1 )  "*"         "*"    "*"
## 8  ( 1 )  "*"         "*"    "*"
## 9  ( 1 )  "*"         "*"    "*"
## 10  ( 1 ) "*"         "*"    "*"
## 11  ( 1 ) "*"         "*"    "*"
## 12  ( 1 ) "*"         "*"    "*"
## 13  ( 1 ) "*"         "*"    "*"
## 14  ( 1 ) "*"         "*"    "*"
## 15  ( 1 ) "*"         "*"    "*"
## 16  ( 1 ) "*"         "*"    "*"
## 17  ( 1 ) "*"         "*"    "*"
```

The relationship seems to be non-linear

Question 7.11

a)

```
set.seed(100)
y <- rnorm(100)
x1 <- rnorm(100)
x2 <- rnorm(100)
```

b)

```
b1 <- 1
```

c)

```
a <- y - b1*x1
b2 <- lm(a~x2)$coef[2]
```

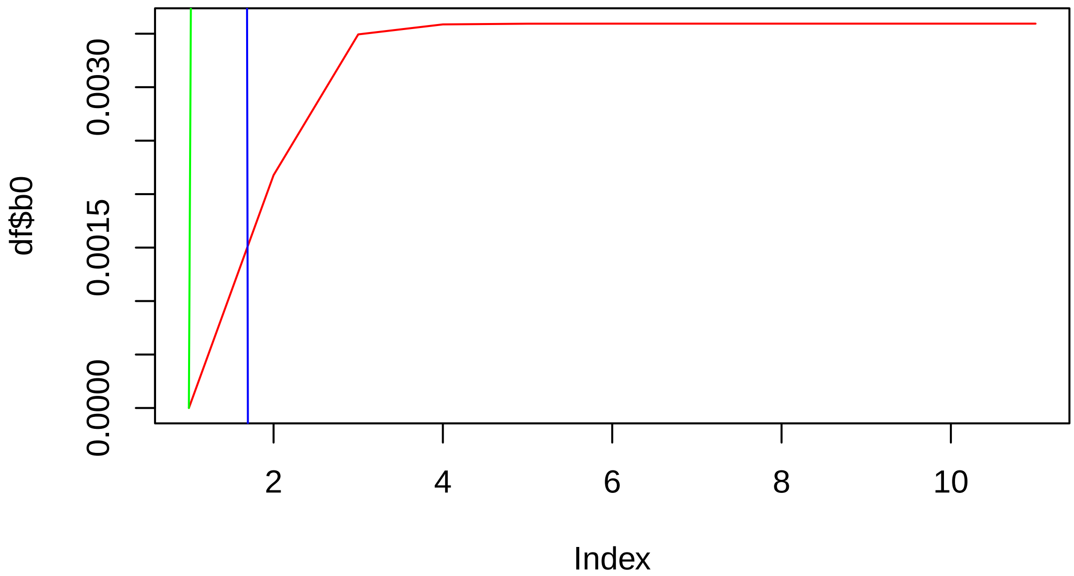
d)

```
a <- y- b2*x2
b1 <- lm(a~x1)$coef[2]
```

e)

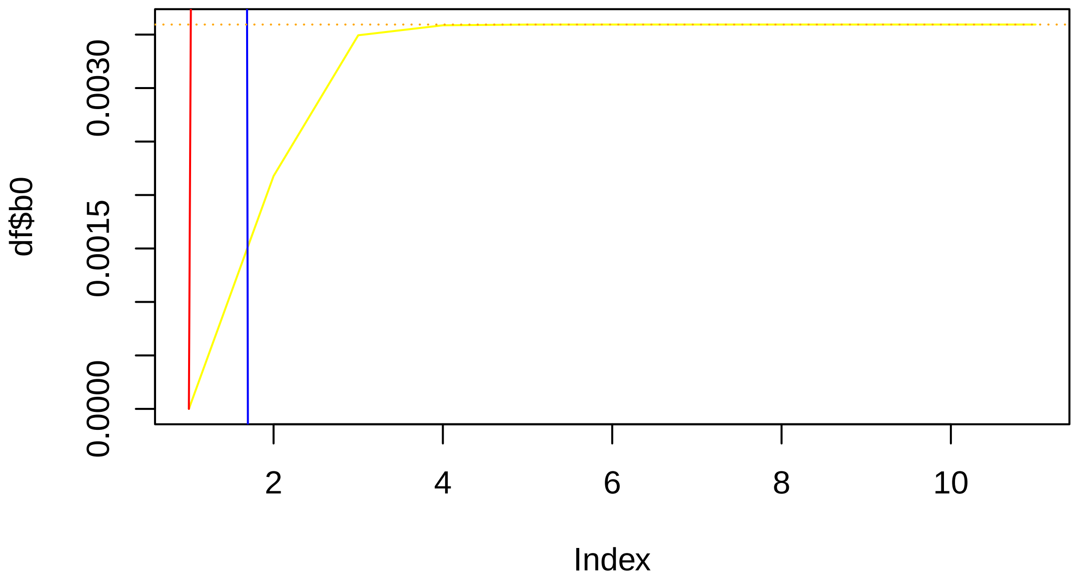
```
iterations <- 10
df <- data.frame(0.0, 0.27, 0.0)
names(df) <- c('b0', 'b1', 'b2')
for (i in 1:iterations) {
  b1 <- df[nrow(df), 2]
  a <- y - b1 * x1
  b2 <- lm(a ~ x2)$coef[2]
  a <- y - b2 * x2
  b1 <- lm(a ~ x1)$coef[2]
  b0 <- lm(a ~ x1)$coef[1]
  b0
  b1
  b2
  df[nrow(df) + 1,] <- list(b0, b1, b2)
}
```

```
plot(df$b0, col = 'red', type = 'l')
lines(df$b1, col = 'blue')
lines(df$b2, col = 'green')
```



f)

```r
plot(df$b0, col = 'yellow', type = 'l')
lines(df$b1, col = 'blue')
lines(df$b2, col = 'red')

d <- coef(lm(y ~ x1 + x2))
abline(h = d[1], col = 'orange', lty = 3)
abline(h = d[2], col = 'purple', lty = 3)
abline(h = d[3], col = 'pink', lty = 3)
```



g)

More than 5 iterations were required for a good approximation.